AMENDMENTS TO THE SPECIFICATION:

Please amend the specification as follows:

Please **amend** the paragraphs previously inserted before line 24 on page 1 of the specification via a Preliminary Amendment filed on March 31, 2005:

Figure 1 shows a flow diagram of an exemplary method for validating computer code, consistent with an embodiment of the present invention[[.]]; and

Figure 2 shows an exemplary structure of a computer program, consistent with an embodiment of the present invention.

Please **amend** the paragraph beginning on page 1, line 20 that was previously amended via a Preliminary Amendment filed on March 31, 2005 as follows:

Objects, aspects and advantages of the invention will be better understood from Figure 1, Figure 2, and the following detailed description of embodiments of the invention.

Please **amend** the paragraphs previously inserted before line 35 on page 4 of the specification via a Preliminary Amendment filed on March 31, 2005:

Figure 1 shows a flow diagram of an exemplary method for validating computer code, consistent with the above-described embodiment and features of the present inventions. In step 110, the program defines a set of definition instructions. The set of definition instructions may be, for example, classes including a class associated with a provider class, the class may include an object with a runtime function. In step 120, the program defines a set of implementation instructions. The set of implementation instructions may be, for example, interfaces having a method. Next, in step 130, the program defines a script code section. In step 140, a validation tool validates the set of definition instructions and the set of implementation instructions by determining whether the class is in compliance with the interface. In step 150, the program determines, during compilation, whether the method of the interface can be used to execute the runtime function. In step 160, the program determines whether the object is relying on a feature that is not promised by the class. In step 170, the program determines whether the interface includes an additional method that may be called during runtime execution of the program. In step 180, the program verifies whether implementation of the provider class is in compliance with a promise of the provider class. In step 190, a symbol table is generated based on the script code section. In step 195, the script code section is validated ~~using~~ by comparing the set of implementation instructions with the symbol table.

Please **insert** the following new paragraph after the paragraph previously inserted before line 35 on page 4 of the specification via a Preliminary Amendment filed on March 31, 2005:

Figure 2 shows an exemplary structure of a computer code 200. The computer code includes definition instructions 202, implementations instructions 204, and a script code section 206. Definition instructions 202 may include a class 208 that may be associated with a provider class 210. Class 208 may include a object 212 and a feature 214. Object 212 may include runtime function 216 and provider class 210 may include a promise 218. Implementation instructions 204 may include an interface 220. Interface 220 may include a method 222 and an additional method 224. A symbol table 226 may be generated by computer program 200 based on script code section 206. Symbol table 226 may include a variable 228.